

Cockpit Interface for Locomotion and Manipulation Control of the NASA Valkyrie Humanoid in Virtual Reality (VR)

Steven Jens Jorgensen¹, Murphy Wonsick^{1,2}, Mark Paterson^{1,3},
Andrew Watson^{1,4}, Ian Chase^{1,4}, and Joshua S. Mehling¹

Abstract—A virtual reality (VR) interface is presented for controlling NASA’s Valkyrie humanoid robot with flexible locomotion control options and intuitive teleoperation. Locomotion modes include navigating to a specified waypoint, sending desired velocities with a joystick, or manually placing a sequence of footsteps. On the other hand, teleoperation modes include voice commands to toggle relative whole-body tracking or high-level commanding, and key bindings for common stored poses such as power grasps. The primary novelties of the interface are in the implementation of the cockpit mode and a floating augmented reality (AR) screen fixed with respect to the robot. The former enables embodied teleoperation and increased awareness during navigation in a mixed-reality setting. The latter preserves depth perception without the disparity clutter from a stereo point cloud. The interface is demonstrated on two real humanoids performing common explosive ordnance disposal (EOD) tasks such as door opening, vehicle inspection, and disruptor placement. Notably, the interface enables a single operator to control multiple humanoids. While the interface is motivated by EOD missions, the presented ideas are usable for other robots employing VR-based control.*

I. INTRODUCTION

As legged robots become more ubiquitous, the Department of Defense (DoD) has been looking to incorporate quadrupeds [1] and humanoids [2] for explosive ordnance disposal (EOD) missions. While humanoids can have more potential than quadrupeds for improvised explosive device (IED) response, existing human-to-humanoid interfaces using a mouse and a keyboard for EOD missions are low in efficiency with the robot being idle up to 50% of the time as the robot waits for operator commands [2]. Notably, similar mouse-and-keyboard interfaces for controlling humanoids [3], [4], [5], [6] were largely developed with the assumption that the mission scenario will have high-latency or poor network connectivity due to its developmental history for nuclear disaster response [7]. These interfaces also had expectations for highly capable planners for long duration autonomy which remain difficult to implement [8].

However, low-latency and reliable communication conditions are possible in certain EOD scenarios, where robots and EOD equipment can be deployed within a few hundred meters away from danger while maintaining line-of-sight [9].

The authors are with the ¹NASA Johnson Space Center, ²Northeastern University, ³METECS, and ⁴Jacobs Technology. This work was supported by the Irregular Warfare Technical Support Directorate (IWTSD).

*Disclaimer: Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration

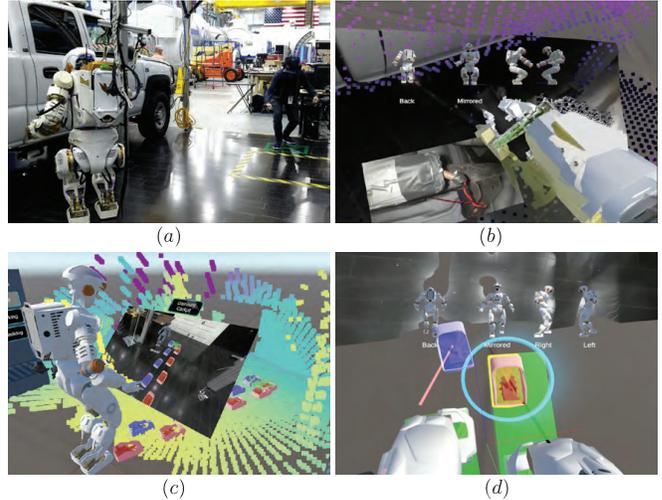


Fig. 1. (a) Valkyrie following the operator’s movements to perform a vehicle inspection with a boom camera. (b) The operator’s point-of-view (PoV) in virtual reality (VR) when the cockpit mode is enabled showing external views of the robot configuration in the heads-up-display (HUD), a floating image of the boom camera attached to the Valve Index controller, RGB and LIDAR point cloud, and two 3D models of the robot’s right arm (white for current config, yellow for desired config). (c) A third person view of the VR interface showing mixed-reality (MR) with a floating augmented reality (AR) screen, LIDAR data, robot model, and planned footsteps from a given waypoint goal. (d) Operator PoV for placing modifiable manual footsteps on the floor where the green region indicates kinematically feasible regions and the blue hoop around the step indicate a viable user command.

Some scenarios can also afford the option of attaching a network tether cable to the robot which effectively removes latency concerns. For these low-latency mission scenarios, a user-interface (UI) that takes advantage of higher throughput semi-supervised control approaches in the form of joystick-based navigation [10], and whole-body teleoperation [11], [12] would be better suited as direct manipulation increases human-to-robot control efficiency [13]. Additionally, this UI can still incorporate more intelligence in the form of waypoint navigation [14] for locomotion and affordance templates for manipulation [15].

For efficient human-to-humanoid control and with guidance from EOD personnel, this paper presents a novel virtual reality (VR) based UI to perform locomotion and manipulation control of the NASA Valkyrie robot [16]. The primary contributions of this work are in the novelty of the UI (Sec. V) particularly in the implementation of the cockpit mode (Sec. V-C), the floating augmented reality (AR) screen (Sec. V-D), the ability to control multiple humanoids

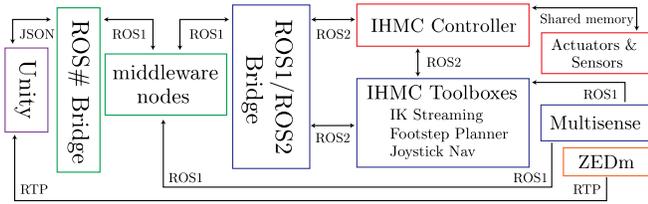


Fig. 2. A high-level architecture diagram of the VR interface. Arrow labels indicate the communication protocol between each module. Unique box colors (red, green, blue, orange, and purple) each indicate a dedicated computer to process modules of the same color. While the Unity module runs on Windows 10, all other modules run on Linux. Only the controller, actuator, and sensor modules are run on the same real-time computer. Finally, the middleware, ROS# bridge, and Unity modules are off-board.

(Sec. VI-B), and the overall user interaction with sensor data in a mixed-reality (MR) [17], [18] setting (Fig. 1).

II. RELATED WORK

State-of-the-art teleoperation for legged robots have been nicely summarized in a recent ICRA workshop [19] with VR-based interfaces being a prominent approach. These works focus on whole-body re-targeting [11], [12], [20], along with defining multi-contact behaviors [21] in third-person VR, and an immersive teleoperated VR approach with an accompanying exoskeleton cockpit called TABLIS [22].

Among a number of VR interfaces for robot control [23], only a few are for humanoids and those closest to this paper are the following works: [24], [25], [21]. The referenced works have similarities to this paper in terms of using virtual artifacts [26] (manipulable virtual 3D objects by humans or robots), waypoint-based or joystick-based navigation, and presentation of data with either environment or user-anchored interface elements. Our work extends these UI implementations with the cockpit mode and the AR screen while also showing a complete hardware demonstration with multiple humanoids. A similar idea to the cockpit mode was explored in [27] which used a homunculus frame of mind bringing the user to a “VR control room” for operating the robot. But, this creates a disembodied mode of teleoperation as opposed to our mixed-reality approach. Our collaborators, the Institute for Human and Machine Cognition (IHMC), have also implemented the cockpit mode but without the ability to zoom-out for increased situational awareness [21].

III. HARDWARE

The VR interface is demonstrated on the Valkyrie humanoid platform [16] capable of high-performance torque control [28]. For perception, the robot has a Multisense SL [29] to obtain RGB-D and LIDAR data and a ZEDm [30] sensor to enable mixed-reality [18]. The VR headset is an HTC Vive Pro [31], but compatible headsets [17] with SteamVR [32] have been confirmed to work. Finally, the interface requires two Valve Index Controllers [33] for locomotion and manipulation control (Sec. V). Due to the selection of VR hardware, base stations [34] are required to perform headset and controller pose tracking. Finally, while the interface can have additional trackers [35] for additional

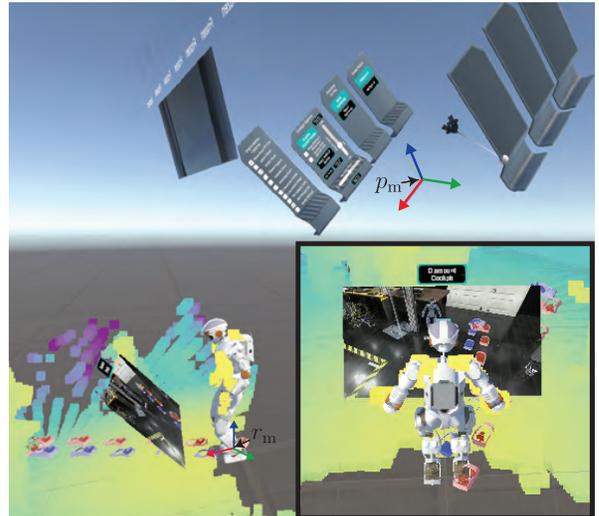


Fig. 3. Zoomed-out view during cockpit mode by placing a fixed transform between the tracker origin frame p_m and robot mid-feet frame, r_m . The figure also shows how the UI buttons have been transformed after following the tracker origin, p_m . The embedded picture shows the user’s PoV.

IK targets, or VR gloves [36] for remapping the user’s fingers’ state to robot finger control, these hardware were not used in the demonstrated EOD use cases (Sec. VI).

IV. ARCHITECTURE

Fig. 2 shows a high-level diagram of the end-to-end architecture of the system. In total there are five computers, three of which are on-board the robot and the last two are off-board. The first on-board computer runs a quadratic-program (QP) based whole-body controller [37] developed by the IHMC with their open robotics software [38]. The controller runs on a real-time thread and communicates via shared memory to send commands to the actuators and receive proprioceptive data from the embedded sensors. The second on-board computer runs the IHMC toolboxes [38], the Multisense SL sensor, and the ROS1/ROS2 bridge [39]. The IHMC toolboxes are used for inverse-kinematics (IK) streaming, footstep planning, and joystick-based navigation and communicates with ROS2 [40] messages. The third on-board computer is an NVIDIA Jetson Xavier [41] which streams the ZEDm video feed directly to the VR interface using RTP [42].

The first off-board computer runs middleware ROS nodes and nodelets which provide services for interfacing with the IHMC controller and toolboxes for commands, filtering ROS point cloud data for bandwidth reduction, and synchronizing timestamped data [43] for VR visualization. The final off-board computer runs the VR interface in Unity [44], a video game engine in Windows 10, which communicates with the middleware nodes via JSON using ROS# for Unity [45], and the ROS bridge suite [46]. Notably, ROS# was modified to enable auto-reconnects when the connection with the ROS bridge suite drops. Finally, the SteamVR plugin [47] is used to interface the VR hardware within Unity.

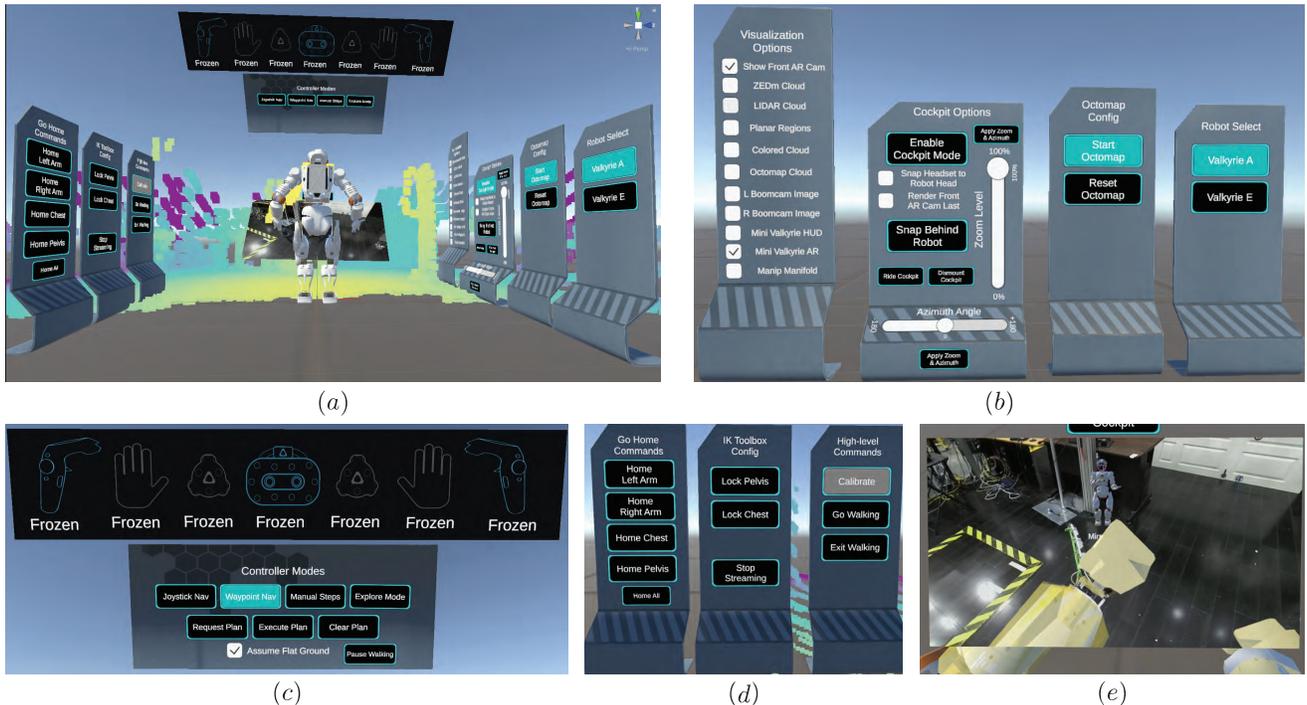


Fig. 4. (a) A third person rear view of the Cockpit-based VR interface showing multiple sensor data such as the robot state, tracker state, point cloud, RGB images, and user options. (b) The right side panel showing view, cockpit, and robot swap options. (c) The top panel showing tracker information and navigation modes. (d) The left side panel showing high-level commands such as sending joints to its home position, IK toolbox options, and robot start-up buttons. (e) A 2D Augmented Reality (AR) screen projected from the robot’s head. The screen integrates VR objects such as desired robot configuration from the IK toolbox (yellow robot model), a third person mirrored view of the robot, and robot footstep plans with the real-world.

V. COCKPIT-BASED VR INTERFACE

The essence of the cockpit-based VR interface is threefold. First, the user-interface (UI) elements always follow either the user origin frame (equivalent to the tracker origin frame), the user’s heads-up-display (HUD), or the tracker frames attached to the user’s body. Second, the user is able to either “ride the skin of the robot” or “zoom out of the robot” (Fig. 3). The former enables the user to see the robot’s arms and legs even if these are not in the field-of-view (FoV) of the robot’s cameras while the latter for example is useful in 3D joystick navigation. Third, all the sensor data such as point cloud, RGB camera feeds, and robot state is available to the user and is always in a mixed-reality (MR) setting. Fig. 4 (a) shows a third person rear view of the VR interface.

Note that for the right, top, and left right panels (Fig. 4 (b)-(d)), general interaction with these UI elements involve pointing the laser emitted from the right controller and pressing the right trigger button. Finally, audio feed-back via text-to-speech [48] is provided to the user during major mode changes (e.g. start/end of IK streaming of joint groups, execution of high-level commands, etc.).

A. Data Visualization and Options

Typically, only a subset of visualization data is useful for any given scenario. For instance, during joystick-based navigation a 3D occupancy map is more informative than an RGB point cloud for avoiding 3D obstacles. Additionally, preference can also dictate whether a floating RGB screen

is more useful than an RGB point cloud; historically, EOD operators are more accustomed to the former. Due to this importance, the user can toggle different sensor data visualization options on the right panel (Fig. 4 (b)).

For point cloud data, the user can toggle LIDAR and RGB-D points from the Multisense, RGB-D points from the ZEDm, or 3D occupancy points from Octomap [49]. Notably, our interface can visualize all the point cloud data simultaneously without performance degradation by voxel filtering ROS point clouds to 5mm leaves and pass-through filtering within a 5m radius around the robot [50]. Bandwidth is also reduced by only passing XYZ point cloud channels and only including RGB channels as needed. This filtering approach also saves JSON deserialization time within Unity. On the other hand, the ZEDm streams camera data directly to Unity via RTP. Bandwidth is saved as the stereo processing of the ZEDm is done on Unity enabling lossless RGB-D point cloud data visualization.

The user can also toggle the AR screen (Sec. V-D) and robot joint state data with a floating 3D avatar of the robot. The 3D avatar of the robot provides robot joint configuration information while the user is sharing the same point-of-view (POV) of the robot. This 3D avatar can either be embedded as part of the AR screen (Fig. 4 (e)) or persistently be displayed on the heads-up-display (HUD), see Fig. 1 (b). The latter is typically used when the AR screen is turned off and the operator is viewing perception data with 3D point clouds.

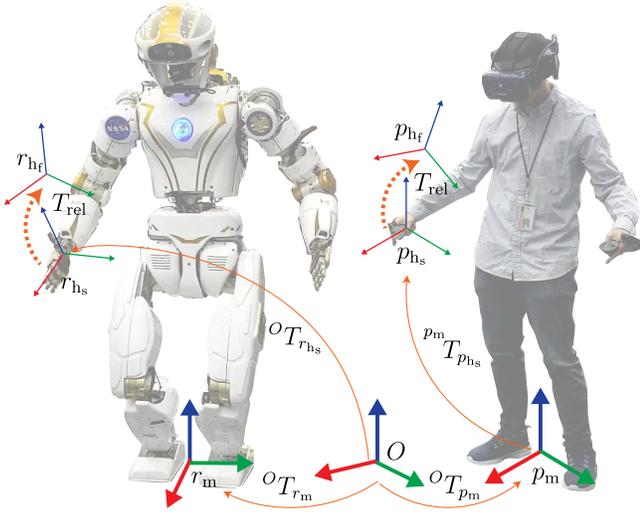


Fig. 5. An illustration of the relative transform, T_{rel} , used to compute the desired right hand Cartesian target pose for the whole-body IK. In the diagram above, T_{rel} is obtained by identifying the 4x4 homogeneous transform to bring the starting tracker pose p_{hs} to the current pose, p_{hf} . The diagram also shows how the user's tracker origin, p_m can be assumed to have the same orientation as the robot's mid-foot pose r_m .

B. Control Modes

In general, the user can send either locomotion (Sec. V-B.1- V-B.3), manipulation (Sec. V-B.4), or high-level commands (Sec. V-B.5). Switching between locomotion modes can be done with the top panel (Fig. 4 (c)). Similarly high-level commands can be sent with the remaining panel options in Fig. 4 (b) and (d). In contrast, whole-body IK streaming is toggled strictly with voice command recognition [51] or an external ROS command.

1) *Waypoint Navigation*: the user can cast a waypoint by holding the left trigger down and change its heading with the analog sticks. When available, the waypoint snaps to detected planar regions in the VR world. Releasing the trigger finalizes the cast. The user can then press the request plan button to receive a plan from the IHMC footstep planner toolbox. If a plan exists, a preview of the footsteps becomes visible to the user in a mixed reality setting (Fig. 1 (a)). The user can then observe the suggested plan, request a new plan, execute the plan, or clear the plan by pressing on sub-panel buttons that become available on the top panel when waypoint navigation is active. Executing the plan initiates the walking sequence and during execution, the pause walking button is available to stop the robot.

2) *Manual Footsteps*: Similar to casting waypoints, the user can cast a left or right footstep step using the left and right trigger buttons respectively as well as change the step yaw with the analog sticks. Visual cues indicate kinematic feasibility and if the user command would be accepted (Fig. 1 (c)). Options include immediate footstep execution upon finalizing a cast, or placing all custom footsteps first, inspecting them, and modifying the steps before execution.

3) *Joystick-based Navigation*: The user can press the left trigger to initiate or stop robot walking. The left analog

stick dictates linear velocity commands for forward, strafing, and backwards walking, while the right analog stick dictates rotational velocity commands. These velocity commands are low-pass filtered then sent to the IHMC joystick navigation toolbox. Footstep plans are received from the toolbox and are dynamically shown to the user in VR.

4) *Whole-body IK Streaming*: Voice commands [51] are used to initiate whole-body tracking of the operator with the keywords “thaw” or “freeze” followed by the robot joint groups to perform tracking. For partial body tracking, uttering “thaw left arm” initiates tracking only on the left arm and “freeze left arm” stops tracking. Whole-body tracking can also be accomplished with “thaw everything” and “freeze everything” which respectively start/stop tracking. The user can check the states of the tracker with the top panel (Fig. 4 (c)). If the base stations lose sight of a tracker, the corresponding joint group is automatically frozen for safety.

The user tracker inputs come from the VR headset for the desired 3 degrees-of-freedom (DoF) head orientation, and the left and right controllers for 6-DoF Cartesian pose. These 15 DoF Cartesian tasks are sent to the IHMC IK streaming toolbox which performs the velocity-level whole-body IK with joint limits, self-collision constraints, jerk smoothing, future tracker pose estimation, etc. The details of the whole-body IK is out of this paper's scope, but the reader is referred to the open-sourced implementation at [38]. While the toolbox can accept more trackers (e.g. torso, pelvis, or elbow trackers), our experiments so far have shown that the headset and two controllers are sufficient for most upper-body humanoid manipulation. This minimum set of trackers also avoid the explicit need for whole-body re-targeting with multiple trackers [11].

Lastly, the IK Cartesian targets for the robot's end-effectors are computed using the relative transform, T_{rel} between the initial tracker pose and the current tracker pose (Fig. 5). Concretely, suppose the initial thawed pose of the right hand controller is p_{hs} and its current pose to be p_{hf} , the homogeneous transform for T_{rel} is computed to be

$$\begin{aligned} T_{\text{rel}} &= {}^{p_m}T_{p_{\text{hs}}}^{-1} {}^{p_m}T_{p_{\text{hf}}} \\ &= {}^{p_{\text{hs}}}T_{p_{\text{hf}}}, \end{aligned} \quad (1)$$

where ${}^{p_m}T_{p_{\text{hs}}}$ and ${}^{p_m}T_{p_{\text{hf}}}$ are the homogeneous transforms of the initial and current poses of the right hand controller respectively expressed in the tracker origin frame, p_m .

Using a relative transform is arguably safer than an absolute transform as all tracking initial conditions would have zero desired velocity.¹ Operationally, it is also intuitive that enabling IK tracking of a group joint would always be relative to the initial configuration of the tracker even during a zoomed-out view of the robot. The relative transform also allows the user to operate in a comfortable workspace by re-thawing and making incremental changes as desired.

¹An absolute desired transform (i.e. assuming that p_m is a proxy for r_m , the robot mid-foot frame, and ${}^{r_m}T_{p_{\text{hf}}} = {}^{p_m}T_{p_{\text{hf}}}$) would have non-zero velocities if there's any pose difference between the current and desired end-effector configurations.

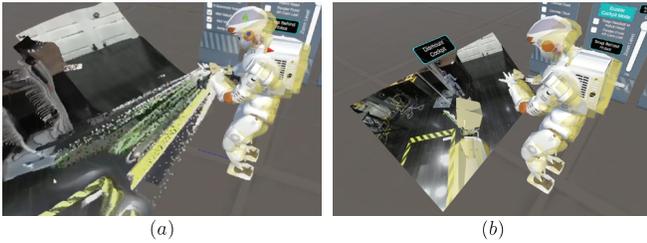


Fig. 6. Sub-figures (a) and (b) show the difference between viewing RGB data with a 3D point cloud versus with an augmented reality (AR) screen. The AR screen has less artifacts as the cloud disparity is not present, but depth-perception is preserved by streaming the stereo data to the user.

5) *High-level commands*: Certain high-level commands such as bringing the robot to default positions (home config), changing the IK toolbox config, or re-initializing the robot is done with UI button presses (Fig. 4 (d)). Other stored poses such as hand power grasps or simple neck movements are done with a controller button press or voice commands.

C. Cockpit Mode

When the user enables the cockpit mode on the right panel (Fig. 4 (b)), a fixed offset transform is applied to p_m from r_m , where p_m is the tracker origin frame and r_m is the mid-feet Z-up frame of the robot (Fig. 5). The frame r_m is computed by taking the average position and yaw of the feet. The fixed offset transform is described in spherical coordinates such that the forward direction of r_m always points to the origin of r_m . Essentially, p_m follows r_m even during robot locomotion. When this fixed offset is set so that $p_m = r_m + \Delta x$, with Δx being an additional fixed position offset to accommodate user height, this enables the user to “ride the skin” of the robot for embodied teleoperation. This lets the user visually match the robot’s pose by placing the controllers to the current pose of the robot’s hands, look at the same direction as the robot before enabling tracking, and observe sensor data outside of the camera’s FoV. Otherwise, the user is able to change the radius, azimuth, and elevation angle of the transform offset for a zoomed-out view (Fig. 3). Due to the cockpit mode and options to view the robot by modifying the offset transform, the user never has to physically step in the real-world nor teleport around in the VR environment to observe the robot’s immediate surroundings. To ensure smooth changes a low pass filter is applied to the offset. Finally, the user can alleviate feeling of nausea by following the sway of the robot’s pelvis.

D. Augmented Reality (AR) Screen

The AR screen is a floating textured 2D sprite showing the RGB data of the ZEDm camera and is projected with the correct horizontal and vertical field-of-view (FoV) angles from the camera’s mounting location (Fig. 6). The left RGB image is only visible to the user’s left eye and the right RGB image is only visible to user’s right eye. In effect, this provides depth perception when the user is looking at the AR screen from the camera mount. The implementation used

here is a significantly modified version of the ZED Unity plugin [52].

Unlike other works where the stereo RGB data is statically placed on the HUD to provide a first-person view [53], [11], [12], our approach in making this screen float in the 3D world by fixing it with respect to the robot model, enables the user to view additional 3D data such as other sources of point cloud and robot joint configuration even with a limited camera FoV. For instance, situational awareness of the biped’s legs is possible despite it not being visible through the camera (Fig. 1 (d)). Importantly, this paradigm is extendable by attaching more floating AR screens around the robot when cameras are available. Similarly, an option to stitch all the camera frames for a higher mixed-reality FoV is also possible. As an additional benefit, the AR screen provides depth perception without additional point cloud disparity artifacts that usually arise from the RGB-D data (Fig. 6). Note that the AR screen is most effective when it is rendered last to prevent virtual 3D objects from being rendered twice from the user’s PoV.

VI. EXAMPLE EOD SCENARIOS

The VR interface is demonstrated with the NASA Valkyrie humanoid on tasks commonly found in urban EOD scenarios: door opening, vehicle inspection, and disruptor placement. The use of two humanoids with only a single operator is also showcased in the latter tasks.

A. Fast door opening and traversal

Figs. 7 (a)-(c) show a door opening and traversal task using the cockpit-based VR interface. The user casts a waypoint in front of the door and the footstep planner [14] plots a path to the waypoint. The robot executes walking while the user “rides” the robot (Fig. 7 (a)). Next, the user performs a door opening motion with whole-body IK streaming on the robot (Fig. 7 (b)). Finally, the user zooms-out of the robot in third person view, increasing situational awareness, inspects the point cloud mapped environment, and sends joystick locomotion commands to strafe the robot through the door (Fig. 7 (c)).

B. Vehicle inspection and disruptor placement with two humanoids

Our interface also enables a single operator to control multiple humanoids by a single button press (see Robot Select panel in Fig. 4 (b)). Figs. 7 (d)-(j) show how our interface is able to control two humanoids to perform a vehicle inspection and disruptor placement task. Fig. 7 (d) shows the user navigating the first robot as it carries a boom camera towards the vehicle. Next, Fig. 7 (e) shows the robot inspecting the car vehicle as the user performs whole-body IK streaming. An improvised explosive device (IED) has been detected using the boom camera as shown in the floating texture attached to the user’s right controller. The user then switches to the second robot to pick up a disruptor with an extension tool (Fig. 7 (f)). The disruptor is a remote-activated device used to forcefully activate an IED, and it is imperative

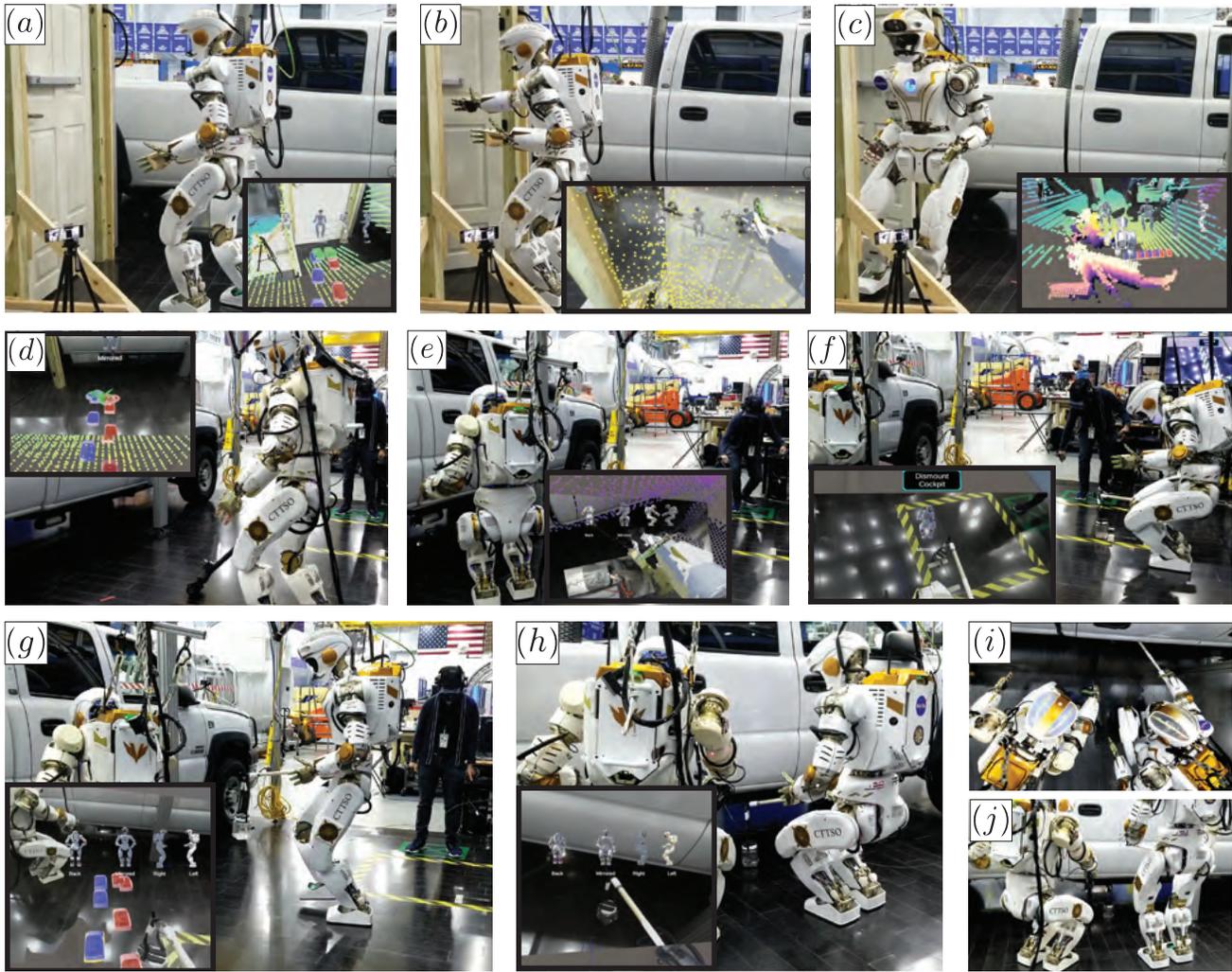


Fig. 7. Subfigures (a)-(c) show Valkyrie performing a door opening and traversal task. With a single operator, sub-figures (d)-(j) show two Valkyrie robots performing a vehicle inspection task and a disruptor placement task. The user point-of-view (PoV) is embedded in sub-figures (a)-(h).

that it is placed as close as possible to the IED. Finally, the user walks the second robot to the IED (Fig. 7 (g)) and places the disruptor beside the IED (Fig. 7 (h)) using the visual aid provided by the first robot. Finally, Fig. 7 (i) shows the robot reaching deep underneath the vehicle and Fig. 7 (j) shows that the disruptor has been successfully placed.

VII. CONCLUDING REMARKS

We have presented a novel VR interface using the concept of a cockpit mode that allows the user to “ride the skin” of the robot or “zoom out” of the robot, enabling increased situational awareness and seamless integration of multi-modal sensor data in a mixed-reality setting. However, some limitations of our work are as follows. As the IK commands are sent in a 1:1 scale, the interface works best with similar human-to-robot proportions. The relative approach to whole-body tracking alleviates some of this limitation, but adding kinematic scaling [12] would better accommodate different users. While it was assumed that the robot operates in a low-latency environment by assuming a network cable exists for communication, the problem of managing this cable was

omitted. Displaying additional feedback to the user without overloading the user with information also remain difficult. It is also arguable what information and interaction elements should surround the user. As such, further user studies with EOD operators are still needed. While these challenges remain, we believe that operating in a mixed-reality setting is pertinent to enable efficient human-humanoid control in VR, and our interface is a step towards that direction.

VIII. ACKNOWLEDGEMENTS

The authors would like to thank our collaborators, IHMC Robotics, for their controller and toolbox implementations which made this UI development possible. Also recognized are the members of the NASA Valkyrie team who supported hardware maintenance during demonstrations, Robert Pierce who worked on an early version of the VR interface, and Michael Steele who ported the Valve Index controllers. We also thank our sponsor, Irregular Warfare Technical Support Directorate (IWTSD), for supporting this work.

REFERENCES

- [1] "Meet Ghost Robotics, the Boston Dynamics of Combat Bots," Washington Post, 2020. [Online]. Available: <https://www.washingtonpost.com/technology/2020/12/07/robot-dogs-patrol-tyndall/> (Last Accessed: September 01, 2021).
- [2] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, et al., "Deploying the NASA Valkyrie humanoid for IED response: An initial approach and evaluation summary," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 1–8.
- [3] H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner, and J. Vice, "Analysis of human-robot interaction at the DARPA robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 3, pp. 420–444, 2015.
- [4] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, et al., "Team IHMC's lessons learned from the DARPA robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.
- [5] M. Johnson, B. Shrewsbury, S. Bertrand, D. Calvert, T. Wu, D. Duran, D. Stephen, N. Mertins, J. Carff, W. Rifenburg, et al., "Team IHMC's lessons learned from the DARPA Robotics Challenge: Finding data in the rubble," *Journal of Field Robotics*, vol. 34, no. 2, pp. 241–261, 2017.
- [6] A. Romay, S. Kohlbrecher, A. Stumpf, O. von Stryk, S. Maniopoulos, H. Kress-Gazit, P. Schillinger, and D. C. Conner, "Collaborative autonomy between high-level behaviors and human operators for remote manipulation tasks using different humanoid robots," *Journal of Field Robotics*, vol. 34, no. 2, pp. 333–358, 2017.
- [7] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski, "The DARPA robotics challenge finals: Results and perspectives," *Journal of Field Robotics*, vol. 34, no. 2, pp. 229–240, 2017.
- [8] C. G. Atkeson, P. B. Benzun, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, et al., "What happened at the DARPA robotics challenge finals," in *The DARPA Robotics Challenge Finals: Humanoid Robots to the Rescue*. Springer, 2018, pp. 667–684.
- [9] D. of Defense, *Explosive Ordnance Disposal (EOD) Operations ATP 4-32*. Army Techniques Publication, 2013, ch. 2: Contingency Operations.
- [10] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami, "An intelligent joystick for biped control," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE, 2006, pp. 860–865.
- [11] K. Darvish, Y. Tirupachuri, G. Romualdi, L. Rapetti, D. Ferigo, F. J. A. Chavez, and D. Pucci, "Whole-body geometric retargeting for humanoid robots," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 679–686.
- [12] M. Elobaid, Y. Hu, G. Romualdi, S. Dafarra, J. Babic, and D. Pucci, "Telexistence and teleoperation for walking humanoid robots," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 1106–1121.
- [13] D. Whitney, E. Rosen, E. Phillips, G. Konidaris, and S. Tellex, "Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality," in *Robotics Research*. Springer, 2020, pp. 335–350.
- [14] R. J. Griffin, G. Wiedebach, S. McCrory, S. Bertrand, I. Lee, and J. Pratt, "Footstep planning for autonomous walking over rough terrain," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 9–16.
- [15] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.
- [16] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, et al., "Valkyrie: NASA's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.
- [17] V. Angelov, E. Petkov, G. Shipkovenski, and T. Kalushkov, "Modern virtual reality headsets," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2020, pp. 1–5.
- [18] M. Speicher, B. D. Hall, and M. Nebeling, "What is mixed reality?" in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–15.
- [19] "ICRA 2021 workshop on teleoperation of dynamic legged robots in real scenarios," IEEE, 2021. [Online]. Available: <https://youtu.be/htM6HW352dc>
- [20] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Bipedal oriented whole body master-slave system for dynamic secured locomotion with LIP safety constraints," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 376–382.
- [21] "Towards humanoid teleoperation of multi-contact maneuvers in constrained spaces," IHMC presentation at IEEE ICRA 2021 workshop on teleoperation of dynamic legged robots in real scenarios, 2021. [Online]. Available: <https://youtu.be/htM6HW352dc?t=30212>
- [22] Y. Ishiguro, T. Makabe, Y. Nagamatsu, Y. Kojio, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit TABLIS," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6419–6426, 2020.
- [23] M. Wonsick and T. Padir, "A systematic review of virtual reality interfaces for controlling and interacting with robots," *Applied Sciences*, vol. 10, no. 24, p. 9051, 2020.
- [24] M. Wonsick and T. Padir, "Human-humanoid robot interaction through virtual reality interfaces," in *2021 IEEE Aerospace Conference (50100)*. IEEE, 2021, pp. 1–7.
- [25] J. Allspaw, G. LeMasurier, and H. Yanco, "Implementing virtual reality for teleoperation of a humanoid robot," *arXiv preprint arXiv:2104.11826*, 2021.
- [26] T. Williams, D. Szafir, and T. Chakraborti, "The reality-virtuality interaction cube," in *Proceedings of the 2nd International Workshop on Virtual, Augmented, and Mixed Reality for HRI*, 2019.
- [27] J. I. Lipton, A. J. Fay, and D. Rus, "Baxter's homunculus: Virtual reality spaces for teleoperation," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 179–186, 2017.
- [28] N. Paine, J. S. Mehling, J. Holley, N. A. Radford, G. Johnson, C.-L. Fok, and L. Sentis, "Actuator control for the NASA-JSC valkyrie humanoid robot: A decoupled dynamics approach for torque control of series elastic robots," *Journal of field robotics*, vol. 32, no. 3, pp. 378–396, 2015.
- [29] "Multisense SL," Carnegie Robotics, 2017.
- [30] "ZED Mini," Stereolabs Inc., 2021. [Online]. Available: <https://www.stereolabs.com/zed-mini/> (Last Accessed: September 01, 2021).
- [31] "Vive Pro," HTC, 2021. [Online]. Available: <https://www.vive.com/us/product/vive-prof> (Last Accessed: September 01, 2021).
- [32] "SteamVR," Valve, 2021. [Online]. Available: <https://www.steamvr.com/en/> (Last Accessed: September 01, 2021).
- [33] "Valve Index Controllers," Valve Corporation, 2021. [Online]. Available: <https://www.valvesoftware.com/en/index/controllers> (Last Accessed: September 01, 2021).
- [34] "Valve base stations," Valve Corporation, 2021. [Online]. Available: <https://www.valvesoftware.com/en/index/base-stations> (Last Accessed: September 01, 2021).
- [35] "Vive tracker," HTC, 2021. [Online]. Available: <https://www.vive.com/us/accessory/vive-tracker/> (Last Accessed: September 01, 2021).
- [36] "Manus gloves," Manus, 2021. [Online]. Available: <https://www.manus-vr.com/> (Last Accessed: September 01, 2021).
- [37] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, and J. Pratt, "Walking on partial footholds including line contacts with the humanoid robot atlas," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1312–1319.
- [38] J. Pratt, P. Neuhaus, D. Stephen, S. Bertrand, D. Calvert, S. McCrory, G. Robert, G. Wiedebach, I. Lee, D. Duran, and J. Carff, "IHMC Open Robotics Software." Institute for Human and Machine Cognition (IHMC), 2021. [Online]. Available: <https://github.com/iuhmrobotics/iuhm-open-robotics-software> (Last Accessed: September 01, 2021).
- [39] "ROS1 Bridge," Open Robotics, 2021. [Online]. Available: https://github.com/ros2/ros1_bridge (Last Accessed: September 01, 2021).
- [40] "ROS2 Foxy," Open Robotics, 2021. [Online]. Available: <https://docs.ros.org/en/foxy/> (Last Accessed: September 01, 2021).
- [41] "Jetson Xavier NX Developer Kit," NVIDIA, 2021. [Online]. Available: <https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit> (Last Accessed: September 01, 2021).
- [42] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC3550: RTP: A transport protocol for real-time applications," 2003.
- [43] J. Faust, V. Pradeep, D. Thomas, J. Perron, M. Carroll, and S. Loretz, "Message Filters," OSRF and Open Robotics, 2021.

- [Online]. Available: http://wiki.ros.org/message_filters (Last Accessed: September 01, 2021).
- [44] “Unity 2019.4.x,” Unity Technologies, 2019. [Online]. Available: <https://unity3d.com/get-unity/download/archive> (Last Accessed: September 01, 2021).
- [45] M. Bischoff, “ROS# v1.6,” Siemens, 2019. [Online]. Available: <https://github.com/siemens/ros-sharp/tree/v1.6> (Last Accessed: September 01, 2021).
- [46] J. Mace, “ROS Bridge Suite,” Robot Web Tools, 2021. [Online]. Available: https://github.com/RobotWebTools/rosbridge_suite (Last Accessed: September 01, 2021).
- [47] “SteamVR Plugin for Unity,” Valve Corporation, 2021. [Online]. Available: <https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647> (Last Accessed: September 01, 2021).
- [48] “UnityWindowsTTS,” Virtuality for Safety, 2019. [Online]. Available: <https://github.com/VirtualityForSafety/UnityWindowsTTS> (Last Accessed: September 01, 2021).
- [49] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Autonomous Robots*, 2013, software available at <http://octomap.github.com>. [Online]. Available: <http://octomap.github.com>
- [50] M. Miknis, R. Davies, P. Plassmann, and A. Ware, “Efficient point cloud pre-processing using the point cloud library,” *Int. J. Image Process.*, vol. 10, no. 2, pp. 63–72, 2016.
- [51] “Voice Input in Unity,” Microsoft and Unity Technologies, 2018. [Online]. Available: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/voice-input-in-unity> (Last Accessed: September 01, 2021).
- [52] “ZED Unity Plugin v3.3,” Stereolabs Inc, 2020. [Online]. Available: <https://github.com/stereolabs/zed-unity/releases/tag/v3.3.0> (Last Accessed: September 01, 2021).
- [53] L. Fritsche, F. Unverzag, J. Peters, and R. Calandra, “First-person teleoperation of a humanoid robot,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 997–1002.